Extended maximum concurrent flow problem with saturated capacity

Congdian Cheng

College of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034, PR China
E-mail: zhiyang918@163.com.

Abstract

The present work studies a kind of Maximum Concurrent Flow Problem, called as Extended Maximum Concurrent Flow Problem with Saturated Capacity. Our major contributions are as follows: (A) Propose the definition of Extensive Maximum Concurrent Flow Problem with Saturated Capacity and prove its solutions exist. (B) Design a approximation algorithm to solve the problem. (C) Propose and prove the complexity and the approximation measures of the algorithm we design.
Keywords: network; concurrent flow; approximation algorithm; approximation measure.
MSC(2000): 90C35(Primary); 90C27 (Secondary)

1. Introduction

The Maximum Concurrent Flow Problem (MCFP) is a main kind of multicommodity flow problems, which was introduced by Matula in 1985, see Shahrokhi and Matula [9], and have been severely studied for more than two decades, see, e.g., [2,4,5,6,9,10]. Motivated by the published literatures of

this research area, we discuss a kind of specific Maximum Concurrent Flow Problem in the present work, termed as Extended Maximum Concurrent Flow Problem with Saturated Capacity (EMCFPSC).

The rest of this article is organized as follows. Some preliminaries are presented in Section 2. Section 3 formulates the problem EMCFPSC and prove the existence of its solutions. Section 4 is specially devoted to designing an algorithm to approximately solve the problem. Section 5 investigates the complexity and precision of the algorithm. Finally, the paper is concluded with Section 6.

2. Preliminaries

This section provides some preliminaries for our sequel research.

A graph $G = (V, E)$ is called as a hybrid graph if its edge can be either directed edge or undirected edge, of which the directed graph and undirected graph are specific cases. Here V and E represent all the nodes (or vertices) and edges (or arcs) of G respectively.

Given graph $G = (V, E)$ with weight (or capacity) c and $H = \{[s_i, t_i] : s_i, t_i \in V; i = 1, 2, \cdots, k\}$, where $c : E \longrightarrow R_+$ and $R_+ = [0, \infty)$, and s_i and t_i express the source and the terminal of commodity i respectively, we call the triad (G, c, H) as a multicommodity network. Let $[s_i, t_i] \in H$; $s_i, v_1, v_2, \cdots, v_l, t_i$ be some nodes of G and be different from each other except for $s_i = t_i$; e_j be the edge of G with endpoint v_j resp. v_{j+1} for $0 \leq j \leq l$, where $v_0 = s_i$ and $v_{l+1} = t_i$. (When e_j is directed, the edges v_j and v_{j+1} must be the original endpoint and the terminal endpoint, respectively.) Then we call $P = [(s_i, e_0, v_1), (v_1, e_1, v_2), \cdots, (v_l, e_l, t_i)]$ as an $s_i - t_i-$ path of (G, c, H).

(To simplify in notation afterwards, we denote (v_j, e_j, v_{j+1}) as (v_j, v_{j+1}) when e_j needn't be indicated; and denote (v_j, e_j, v_{j+1}) as e_j when v_j and v_{j+1} needn't be indicated.) Let $\mathcal{P}_i$ be a set of $s_i - t_i-$ paths such that $c(e) > 0$ for all $e \in E(P)$ and $P \in \mathcal{P}_i$, $i = 1, 2, \cdots, k$, where $E(P)$ denotes the set of all the edges of P. We call $\mathcal{P} = \bigcup\limits_{i=1}^{k} \mathcal{P}_i$ as a (positive) path system on (G, c, H), which is denoted by $\mathcal{P}|_{(G,c,H)}$, or $\mathcal{P}$ for conciseness.

Definition 1. Let $\mathcal{P}$ be a path system. If mapping $y : \mathcal{P} \longrightarrow R_+$ satisfies:

$$\sum_{e \in E(P), P \in \mathcal{P}} y(P) \leq c(e) \quad \forall e \in E(\mathcal{P}),$$

where $E(\mathcal{P}) = \{e \in E(P) : P \in \mathcal{P}\}$, i.e. the set of all the edges of $\mathcal{P}$. Then we call y as a multicommodity flow on $\mathcal{P}$ defined by the function on the path system, and as a flow for simplicity; and call $V(y) = \sum\limits_{P \in \mathcal{P}} y(P)$ as flow value of y, and $V_i(y) = \sum\limits_{P \in \mathcal{P}_i} y(P)$ as branch flow value of commodity i with y. The set of all the flows on $\mathcal{P}$ is denoted by $F[\mathcal{P}]$, and by F in brief.

Remark 1. As is known to us, there are two kinds of definitions about the flow, which are the defined by the function on the edge set and by the function on the path system. For the purpose of being simple and clear, we only make the study as far as the flow defined by the function on the path system in the present work.

Generally, the problem to find a multicommodity flow to satisfy certain conditions is called the multicommodity flow problem (MFP). A few of usual multicommodity flow problems are as follows. The problem to find a flow y so that $V(y)$ is maximized, i.e. $V(y) = \mathrm{OPT}[\mathcal{P}] = \hat{V}(= max\{V(y) : y \in F[P]\})$, is called the maximum multicommodity flow problem on $\mathcal{P}$ (MMFP). A flow y satisfying MMFP is called a solution of MMFP. The problem to find

a flow y so that $V_i(y) \le b_i$ for given b_i, $i = 1, 2, \cdots, k$, and $V(y)$ is maximized, i.e. $V(y) = \overline{V}(= max\{V(y) : V_i(y) \le b_i, i = 1, 2, \cdots, k; y \in F\})$, is called the maximum flow problem with bounds(MMFP-B). The problem to find a flow y so that $V(y)$ is maximized under the condition $V_i(y) = \lambda b_i$ for given b_i, $i = 1, 2, \cdots, k$, and for $\lambda \in [0, 1]$ is called the Maximum Concurrent Flow Problem (MCFP) (see, e.g., Shahrokhi and Matula [10], or Garg and Könemann [6]).

3. Model formulation

Now we begin to introduce the problem we are attacking in the present work.

Definition 2. Suppose $\mathcal{P}$ is a path system on the multicommodity network (G, c, H), and $\mathbf{b} = (b_1, b_2, \cdots, b_k)$. We call the problem to find a flow y' so that $\min\limits_{1\le i\le k}[\frac{1}{b_i}V_i(y')] = \max\{\min\limits_{1\le i\le k}[\frac{1}{b_i}V_i(y)] : y \in F, V_i(y) \le b_i\}$ as Extended Maximum Concurrent Flow Problem (EMCFP). We call the problem to find a flow y' so that $\min\limits_{1\le i\le k}[\frac{1}{b_i}V_i(y')] = \max\{\min\limits_{1\le i\le k}[\frac{1}{b_i}V_i(y)] : y \in F, V_i(y) \le b_i\}$, and $V(y') = \max\{V(y) : y$ is the solution of EMCFP $\}$ as Extended Maximum Concurrent Flow Problem with Saturated Capacity(EMCFPSC).

Theorem 1. The solution of the problem EMCFPSC exists.

Proof. It is obvious that the solution of MCFP is also the solution of EMCFP. Hence, $B = \{y \in F : y$ is the solution of $FMCFP\} \neq \emptyset$ from the well known fact the solution of the problem MCFP exists. When the member of B is finite, the conclusion of Theorem 1 is trivial. Otherwise, $\sup\{V(y) : y \in B\}$ exists and is finite. Let it be b. Then there is at least a sequence $\{y_j\}$ in B such that $\lim\limits_{j\to\infty}[\sum\limits_{i=1}^{k} V(y) = b$, and $\lim\limits_{j\to\infty} y_j(P)$ exists for all $P \in \mathcal{P}$. Now let

$y(P) = \lim_{j\to\infty} y_j(P)$ for all $P \in \mathcal{P}$. Then $y =\in B$, and $V(y) = b$. That is, y is the solution of EMCFPSC. $\square$

Example During the winter between the year 2009 and 2010, in major regions of the world, the amount of energy consumed for heating is far more than the ordinary winter due to the abnormal lower air temperatures. With this circumstances, the electrical networks emerge to congestion for a lot of districts in the world. Assume the purpose of supplying power is firstly to maximize the minimum of the met rates of cities. Then the problem to optimize the power supplement can be largely regarded as the mould EMCFPSC. One can believe the mean of applications of the mould EMCFPSC from the instance.

4. Algorithm

In this section, we are devoted to designing an approximation algorithm for the problem EMCFPSC.

Algorithm (Approximation algorithm for EMCFPSC):

Input Multicommodity network (G, c, H), path system $\mathcal{P}$, vector $\mathbf{b}$ and error parameter $0 < \eta < 1$.

Output Flow y^* on $\mathcal{P}$, which has the characters specified by the Theorem in Section 5.

1. Put $\epsilon = \min\{\frac{\eta}{\sum_{i=1}^{k} b_i}, \frac{1}{2}\}; b_i^0 = b_i, i = 1, 2, \cdots, k; l = 0, h = 0.$
2. Set $l = l+1$. If $l\eta > 1$, implement the next Step. Otherwise, put $l = l^*$ and go to the Step 4.
3. Set $b_i = l\eta b_i^0, \mathbf{b} = (b_1, b_2, \cdots, b_n)$. For $\mathcal{P}, \mathbf{b}$ and ϵ, obtain an approximation solution y_1 of MMFP-B with by Algorithm 2 of Cheng [3]. If

$V(y_1) < \frac{1}{1+\epsilon}(\sum\limits_{i=1}^{k} b_i)$, put $l = l^*$ and implement the next step. Otherwise, return to (2).

4. Construct the auxiliary network and demanding vector $\mathbf{b}$ as follows. Set $V' = V \bigcup \{t'_0, t'_i : i = 1, 2, \cdots, k\}(t'_0, t'_i \bar{\in} V), E' = E \bigcup \{(t_i, t'_0), (t_i, t'_i) : i = 1, 2, \cdots, k\}, H = \{[s_i, t'_i], [s_i, t'_0] : i = 1, 2, \cdots, k\}, G' = (V', E'); \mathcal{P}'_i = \{P + (t_i, t'_i) : P \in \mathcal{P}_i\}, \mathcal{P}'^i_0 = \{P + (t_i, t'_0) : P \in \mathcal{P}_i\}, i = 1, 2, \cdots, k; \mathcal{P}' = (\bigcup\limits_{i=1}^{k} \mathcal{P}'_i) \bigcup (\bigcup\limits_{i=1}^{k} \mathcal{P}'^i_0); b'_i = (l^* - 1)\eta b^0_i, b'_0 = b, \mathbf{b}' = (b'_1, b'_2, \cdots, b'_k, b'_0)$, and

$$c'(e) = \begin{cases} c(e) & e \in E(G) \\ b'_i & e = (t_i, t'_i) \\ b_i - b'_i & e = (t_i, t^0_i) \end{cases} .$$

5. Put $h := h + 1, b = h\eta$. If $b > \sum\limits_{i=1}^{k} b^0_i$, put $h^* = h$ and implement the next step. Otherwise for $\mathcal{P}', b'$ and ϵ, obtain an approximation solution y_2 by Algorithm 2 of Cheng [3]. If $V(y_2) < \frac{1}{1+\epsilon}[(\sum\limits_{i=1}^{k} b'_i) + b]$, put $h^* = h$ and implement the next step. Otherwise, put $y' = y_2$, and then return to 5.

6. Put $y^*(P) = y'(P + (t_i, t'_i)) + y'(P + (t_i, t_0)), \forall P \in \mathcal{P}_i, i = 1, 2, \cdots, k$. Stop.

5. Algorithm analysis

This section is specially devoted to discussing the correctness, approximate precision and complexity of the algorithms we nave presented above.

Theorem The complexity of the Algorithm is $O(\frac{1}{\eta^3} km^2 \log n)$, where $k = |H|, n = |V(G)|, m = |E(G)|$. Let y be a solution of EMCFPSC and y^*

be the output of the Algorithm. Then y^* is a flow on $\mathcal{P}$, $V_i(y^*) \leq b_i, i = 1, 2, \cdots, k, V(y) \leq [l^*(\sum\limits_{i=1}^{k} b_i) + h^*]\eta$, and

$$[(l^*-1)(\sum_{i=1}^{k} b_i) + (h^*-1)]\eta - 2\eta \leq V(y^*) \leq [(l^*-1)(\sum_{i=1}^{k} b_i) + h^*]\eta.$$

$$(l^*-1)\eta - \frac{2\eta}{\min b_i} \leq \min \frac{V_i(y^*)}{b_i} \leq \min \frac{V_i(y)}{b_i} \leq l^*\eta.$$

Proof. It is obvious that the complexity of the Algorithm depends on the complexity of the subroutine Algorithm 2 of [7] and the times of the iteration that we operate on the subroutine. By simply analyzing the Algorithm, we can know that the times is no more than $\max b_i(\frac{k}{\eta} + \frac{1}{\eta})$, which can be denoted as $O(\frac{1}{\eta})$. On the other hand, the complexity of the subroutine is $O(\frac{1}{\epsilon^2} km^2 \log n) = O(\frac{1}{\eta^2} km^2 \log n)$, see [3]. Hence the complexity of Algorithm is $O((\frac{1}{\eta^3} km^2 \log n)$.

By further analyzing the Algorithm, we can easily know that y^* is a flow on $\mathcal{P}$, $V_i(y^*) \leq b_i, i = 1, 2, \cdots, k$, $V(y) \leq [l^*(\sum\limits_{i=1}^{k} b_i) + h^*]\eta, V(y^*) \leq [(l^*-1)(\sum\limits_{i=1}^{k} b_i) + h^*]\eta$, and $\min \frac{V_i(y^*)}{b_i} \leq \min \frac{V_i(y)}{b_i} \leq l^*\eta$. Hence we only specify $[(l^*-1)(\sum\limits_{i=1}^{k} b_i) + (h^*-1)]\eta - 2\eta \leq V(y^*)$ and $\min \frac{V_i(y^*)}{b_i} \geq (l^*-1)\eta - \frac{2\eta}{\min b_i}$.

In terms of the Algorithm, we have $\epsilon \leq \frac{\eta}{\sum\limits_{i=1}^{k} b_i}, (l^*-1)\eta \leq 1, (h^*-1)\eta \leq$

$\sum_{i=1}^{k} b_i$, and $V(y^*) \geq \frac{1}{1+\epsilon}[(\sum_{i=1}^{k}(l^*-1)\eta b_i) + (h^*-1)\eta]$. Therefore

$$\begin{aligned} V(y^*) &\geq \tfrac{1}{1+\epsilon}[(\sum_{i=1}^{k}(l^*-1)\eta b_i) + (h^*-1)\eta] \\ &= [(l^*-1)(\sum_{i=1}^{k} b_i) + (h^*-1)]\eta - \tfrac{\epsilon}{1+\epsilon}[(l^*-1)\eta(\sum_{i=1}^{k} b_i) \\ &\quad +(h^*-1)\eta] \\ &\geq [(l^*-1)(\sum_{i=1}^{k} b_i) + (h^*-1)]\eta - \epsilon[(\sum_{i=1}^{k} b_i) + (\sum_{i=1}^{k} b_i)] \\ &\geq [(l^*-1)(\sum_{i=1}^{k} b_i) + (h^*-1)]\eta - 2\eta. \end{aligned} \tag{1}$$

Let $V_i' = \sum_{P \in \mathcal{P}_i'} y_i'(P), i = 1, 2, \cdots, k$ and $V_i'^0 = \sum_{P \in \mathcal{P}_i'^0} y_i'^0(P)$. Then $V_i(y^*) = V_i' + V_i'^0$ since $y_i = y_i' + y_i'^0$. As $b_i' = (l^*-1)\eta b_i$, we can have $V_i' \leq (l^*-1)\eta b_i$. Suppose $V_j(y^*) < (l^*-1)\eta b_j - 2\eta$ for some j. Then

$$\begin{aligned} V(y^*) &= \sum V_i = [\sum_{i \neq j}(V_i' + V_i'^0)] + V_j(y^*) \\ &< \sum_{i \neq j} V_i' + \sum_{i \neq j} V_i'^0 + (l^*-1)\eta b_j - 2\eta. \end{aligned} \tag{2}$$

On the other hand, $\sum_{i \neq j} V_i'^0 \leq \sum V_i'^0 \leq (h^*-1)\eta$. Hence, (2) implies

$$\begin{aligned} V(y^*) &< [(\sum_{i \neq j}(l^*-1)\eta b_j) + (h^*-1)\eta] + (l^*-1)\eta b_j - 2\eta \\ &= [(\sum (l^*-1)\eta b_j) + (h^*-1)\eta] - 2\eta. \end{aligned} \tag{3}$$

It is clearly that (3) is in contradiction with (1). So, $V_i(y^*) \geq (l^*-1)\eta b_i - 2\eta$ for any i. This implies

$$\min \frac{V_i(y^*)}{b_i} \geq (l^*-1)\eta - \frac{2\eta}{\min b_i}.$$

This comples the proof. $\square$

Remark 2. Recently, Büsing and Stiller [1] considered a kind of network flow problem arising in line planning. It may be an interesting topic to explore the application of the present work in line planning. Moreover, Soleimani-damaneh [11] investigated the determination of maximal flow in a fuzzy dynamic network with multiple sinks. Mehri [8] studied the inverse maximum dynamic flow problem. We believe to extend the present work in a fuzzy dynamic network and to address the inverse problem EMCFPSC may also be interesting topics.

6. Concluding remarks

In this paper, the problem Extended Maximum Concurrent Flow Problem with Saturated Capacity is introduced. The existence of its solutions is proved. A approximation algorithm to solve the problem is designed, and the effectiveness of the algorithm, including the complexity and approximation measures, is discussed. The approach we design the algorithm is the specific contribution of the present work, whose main characters are to construct auxiliary network and to implement iterating search through subroutine a known algorithm. To solve other network problems with this approach is an interesting topic for further researches in the future.

Acknowledgements The authors cordially thank the anonymous referees for their valuable comments which lead to the improvement of this paper.

This research is Supported by the Eduction Department Foundation of Liaoning Province (L2010514).

References

[1] C.Büsing, S.Stiller. Line Planning, Path Constrained Network Flow and Inapproximability. Networks 57(1) (2011): 106-113.

[2] J.Biswas, D.W.Matual. The two-flow routing algorithms for the maximum concurrent flow problem. Proceedings of the ACM Fall Joint Conference, ACM, New York, 1986, pp. 629-635.

[3] C.D.Cheng, Z.P.Li. Improvements on the proof of an approximate scheme for the maximum multicommodity flow problem. Proceedings of The 4th International Conference on Intelligent Information Technology Application, Vol.II, 2010, pp. 311-314.

[4] S.W.Chiou. A combinatorial approximation algorithm for concurrent flow problem and its application. Comput Oper Res 32(2005): 1007-1035.

[5] L.K.Fleischer, K.D.Wayne. Fast and simple approximation schemes for generalized flow. Math program Ser A 91(2002): 215-238.

[6] N.Garg, J.Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, 1998, pp. 300-309.

[7] B.Korte, J.Vygen. Combinatorial optimization theory and algorithms. Berlin: Springer-Verlag, 2000.

[8] B.Mehri. The inverse maximum dynamic flow problem. SCIENCE IN CHINA Series A 53(2010): 2709-2717.

[9] F.Shahrokhi, D.W.Matula. On solving large maximum concurrent flow problems. Proceedings of the ACM Computer Conference, ACM, New York, 1987, pp. 205-209.

[10] F.Shahrokhi, D.W.Matula. The maximum concurrent flow problem. J Assoc Comput Math 37(1990): 318-334.

[11] M.Soleimani-damaneh. Maximal flow in possibilistic networks. Chaos, Solitons and Fractals 40(2009): 370-375.